



DEVICE TO IMPROVE THE SAFETY OF TAXI USERS AND DRIVERS APPLYING FACE DETECTION

Jesús D. Quintero-Polanco, Andrés F. Charry-Moreno and Oscar M. Torrente Barragán.
Department of Electronic Engineering, Faculty of Engineering, Surcolombiana University, Neiva, Huila, Colombia
E-Mail jdavid@usco.edu.co

ABSTRACT

The prototype implements a system of facial detection developed in Python with OpenCV's framework and a pre-trained convolutional neuronal network to identify the exact moment in which one or more users board the taxi; during this time some pictures of users' and taxi drivers' faces are captured in different amounts of time and are restored in a MySQL data base of a web server. There was a development of a mobile app on Android studio that allows users to observe the taxi driver information, and at the same time, to observe the outline of the route in real time which is also restored in the data base. It was developed a web platform that register or updated taxi drivers' information and inquiry the register of different services developed by the taxi driver in case that there is an unusual situation.

Keywords: API, fotograma, framework, GPS, método, servidor, cámara, internet.

1. INTRODUCTION

Nowadays, Colombia's index of insecurity inside taxis has been increasing significantly for the driver and for the users. Crimes like "paseo millonario", personal objects or vehicles theft, or even murders are the reasons why the society feels scared day by day at the moment of taking this transportation service. Those facts, have generated a nonconformity process in both parts. For example, the taxi drivers decide not to facilitate the service in certain places because they consider them high risk places for their physical integrity, and that is the reason why the users are constantly complaining because of the bad quality service and decide to stop using it (El espectador, 2018).

This leads users to choose to use other transportation means that provide confidence and security during the service, such as Uber, generating a decrease in the use of taxis for individual transport. If users continue abandoning this service, some taxi centers will decrease their profits and may even go to bankrupt and abandon their source of income.

Due to this, there is a necessity to create a prototype device that allows generating a certain degree of trust between both parts, the user and the taxi driver. Thus, they can be more calm when making the respective trip. With the developed prototype device, the user in the event in which he or she is the victim of a crime and does not remember the number of the taxi license plate, as well as the face of the taxi driver, may contact the taxi company. and there, with the information saved, the taxi driver can be identified and started to carry out the necessary investigations to clarify the case. In the same way, if the taxi driver were the victim, the taxi company may have the information and contact the authorities to carry out the process and locate the criminals.

2. METHODOLOGY

The project was developed in four phases, which are described below.

2.1 Phase One Configuration and Development of the Embedded System

An ELP 1080P webcam was used to acquire the image of the interior of the taxi, which was installed on the windshield above the rear-view mirror with a support, since a best range of visibility is obtained to detect the face of the taxi driver and the users.

The image obtained from the camera is captured using a Raspberry pi 4 and processed in Python using the OpenCV library, which is analyzed in different regions of the seats to obtain a better detection of the faces.



Figure-1. Detection regions.

Each region is processed by a pre-trained convolutional neural network of the Caffe framework, which returns the faces that it considers real based on a confidence value (0 to 1) and those that exceed a previously established threshold based on field tests are selected to avoid false positives. The finally detected faces are framed with a box and the confidence value transformed to a percentage delivered by the neural network is displayed.

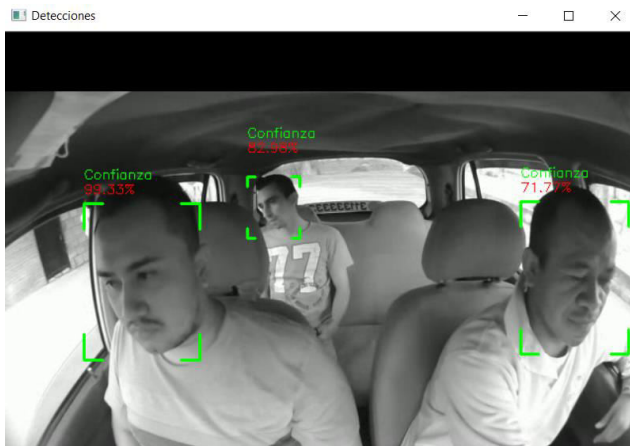


Figure-2. Image with detection frames.

When more than one face is detected, it means that one or more passengers boarded the taxi. In that moment, an http request is sent to the backend of the system to activate the service and images are captured of the faces of users and taxi driver every thirty seconds, those images are stored in a local folder and uploaded to the web server.

When users leave the taxi, the algorithm ends the service and a final image is of the taxi without passengers is captured to obtain a record if a user leaves a personal object abandoned.

To store the photos to the web server, an internet connection is needed, for which the Raspberry board was connected via WiFi with a mobile device that has a 4G connection, which is used to view the App that will be described in the following phases.

2.2 Phase Two. System Backend Development

The backend was developed using the Laravel framework, which is responsible for managing all the http requests that come from the entire system.

The control of the MySQL database which contains all the information of the system is carried out divided into four tables: the users' table contains the information of all the taxi drivers registered by the taxi central with the basic information of the car that they drive.

The routes' table is responsible for saving information on each performed service (route provided by the GPS, service start and end date), the images' table contains information on all the photographs captured during each service. The images table is related to the routes table through an id to link all the information of each service and the routes table. The routes table is responsible for saving information on each service performed (route provided by the GPS, service start and end date), the images table contains information on all the photographs captured during each service. The images table is related to the routes table by means of an id to link all the information of each service. At the same time, the routes table is related to the users table to identify the taxi driver who performed the service.

The users_app table (related to the users table) contains the login information of each taxi driver when entering the mobile application.

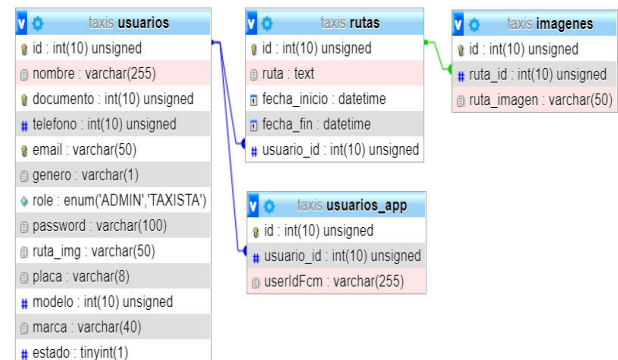


Figure-3. List of database tables.

In Laravel controllers you will find the logic for handling system requests which are listed below:

- **Taxi driver Controller Registration:** handles the requests that come from a form created in HTML with CSS styles to register a new taxi driver quickly, where the data of the taxi driver and taxi are entered, it also contains a button to clean the form and another to register the information in the database.
- **User Controller:** handles the requests that come from the frontend form (Angular) to show, add, create, delete or search for a taxi driver.
- **Route Controller:** handles the requests that come from the frontend to show, search or delete service records and receives requests from the mobile application to refresh or finalize the record of the information received from the GPS.
- **Auth Controller:** contains the logic for the authentication of the system such as the login on the web page and in the mobile application, in the same way it generates a token that identifies each login to provide greater security to the system.
- **Notification Controller:** contains the logic to send activation of the service to the application, upload images of the service, register and delete information about the user logged into the App.



Figure-4. Taxi driver registration form.

Finally, the Laravel routes are established to control all HTTP requests and that is where the addressing that relates to each controller is found to respond to the requests made by the system.

2.3 Phase Three. System Frontend Development

The frontend consists of the development of the graphic part where the central taxi can access to control the system and its respective records. A web page was developed where only people with the role of administrator can enter the information of the system through an email and password.

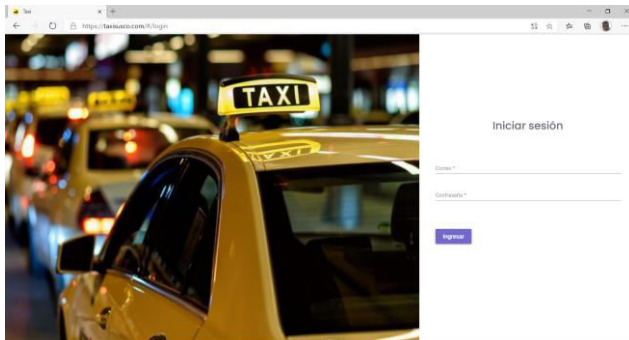


Figure-5. Web page login view.

Once the administrator credentials are confirmed, the administrator enters the main page which has a clean design and can search for the services performed by all the taxi drivers as well as to modify or add records.

In the list of services, you can see the name of the taxi driver, his identification, the license plate of the taxi he is driving and the start and end date of the service. This list can be organized based on the data named above and a general search filtered by the data can be performed as required by the administrator, which facilitates a search when the list of services is extensive.

Servicio	Identificación	Placa	Fecha Inicio	Fecha Fin
10	107293204	B40387	2020-09-29 16:06:01	2020-09-29 16:05:04
10	107293204	B40387	2020-09-29 16:46:16	2020-09-29 16:46:38
10	107293204	B40387	2020-09-29 16:45:38	2020-09-29 16:50:37
10	107293204	B40387	2020-09-29 16:56:04	2020-09-29 17:01:01
10	107293204	B40387	2020-09-29 17:01:04	2020-09-29 17:06:01
10	107293204	B40387	2020-09-29 20:03:23	2020-09-29 20:12:33
10	107293204	B40387	2020-09-29 20:11:38	2020-09-29 20:21:31
10	107293204	B40387	2020-09-29 20:20:19	2020-09-29 20:41:26

Figure-6. Services view web page.

When you want to see more information about a specific service, click on the pencil icon located to the right of the mentioned record and you will be able to view through a Google Maps map the route that the taxi took during the service, identifying the starting point with a GPS icon marker and the arrival point with an icon flag.

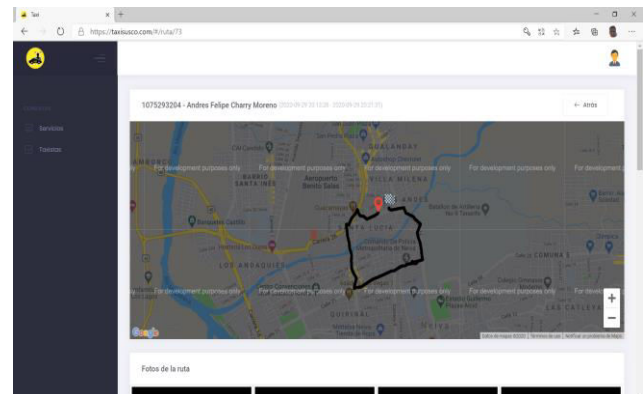


Figure-7. Service information view part 1.

In the same way, when navigating under the map, you can see all the photographs captured during the service where the face of the taxi driver and the passengers are displayed.

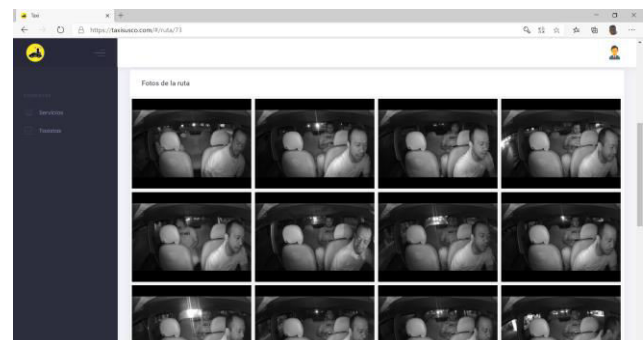


Figure-8. Service information view part 2.

The administrator can also remove a service from the list by clicking on the recycling basket icon located on the right side of the record.

Another important part of the website is the list of taxi drivers where their name, identification and email are



displayed, it is possible to know if they are performing a service (active) or are currently available (inactive).

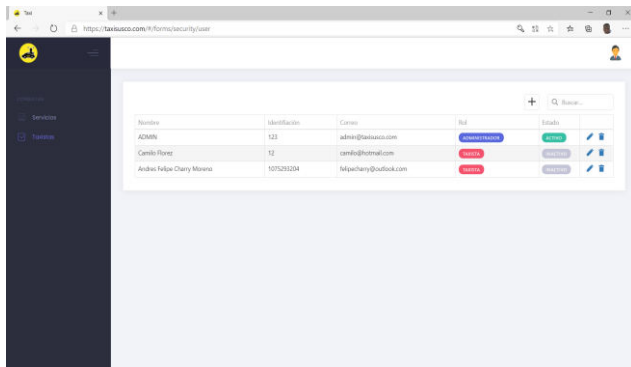


Figure-9. Taxi drivers web page view.

When the administrator needs to update data about a taxi driver, he presses on the pencil icon located to the right of said record and the form containing his current data can be viewed to modify any field and finally save changes.

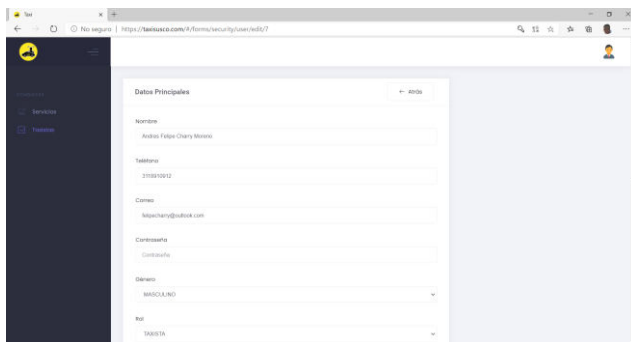


Figure-10. Taxi driver information view web page.

In the same way, a new taxi driver can be registered using a form in which a name, document, telephone number, email, gender, password and a photograph as well as the license plate, model and brand of the taxi that they drive are requested. It is also allowed to delete a taxi driver through the recycling basket icon located in the right of the record.

2.4 Phase Four. Mobile Application Development

Finally, a mobile application was developed through the Android Studio development environment, it allows any registered taxi driver to log in using their document and password.

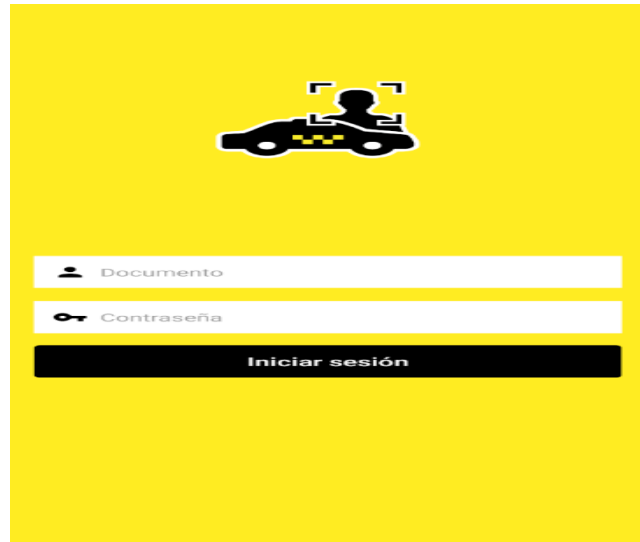


Figure-11. Login app view.

The application was developed with the aim of allowing passengers to view the information of the taxi driver and the taxi that they are boarding, as well as to observe the route that the taxi is taking.

This App is displayed horizontally and on the left you can see some information about the taxi driver such as: photograph, name, email and telephone number, information that can be hidden through a button to better visualize the map if desired.

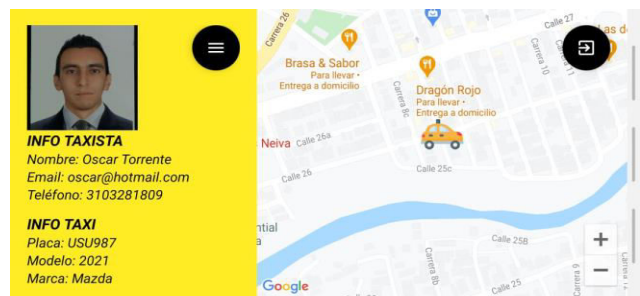


Figure-12. App service view.

In the right part of this information, the Google Maps map is displayed in which the route that the taxi is taking is dynamically observed, where the current position is identified by means of a taxi icon. As the taxi progresses, the route is traced with a polyline by joining the current positions that the GPS has recorded.

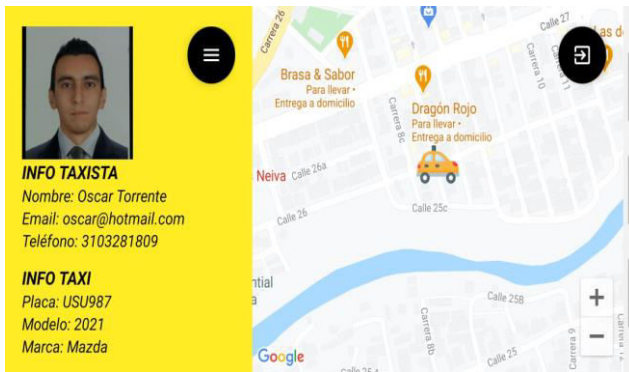


Figure-13. APP route trace visualization.

The application traces the route only when the embedded system using the Python algorithm identifies the existence of passengers by detecting their faces, at that moment the code sends a notification that alerts the app to initialize the route record.

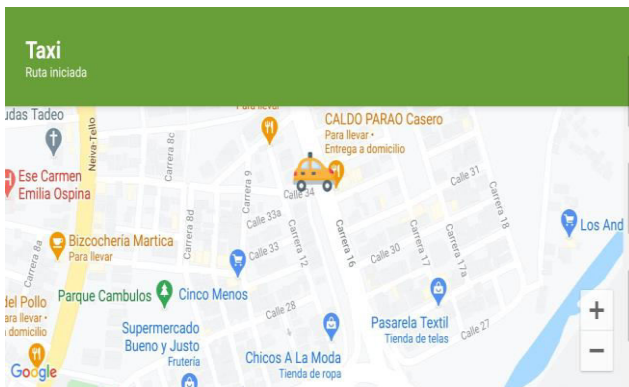


Figure-14. Informational message start of route app.

In the same way, the algorithm makes the application finish the route layout immediately after detecting that the passengers have gotten off the taxi. It is important to mention that the application traces the route even when it is in the background.

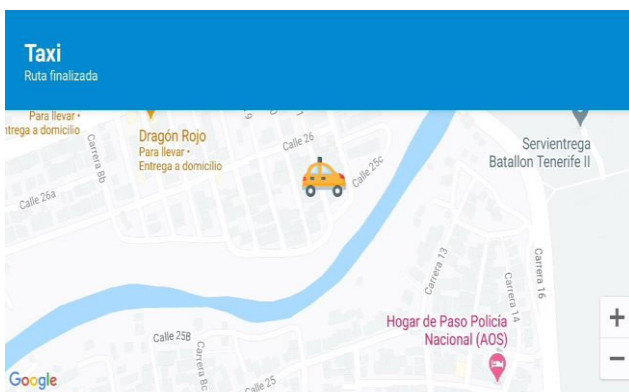


Figure-15. End of route informational message app.

In addition, the application works only when the GPS is turned on, this is confirmed when the taxi driver

logs in and if it is deactivated, an activation request is made so that the application works correctly.



Figure-16. GPS app activation validation.

3. CONCLUSIONS

With this project, it was possible to design and implement a prototype device that significantly improves the safety of taxi users and drivers since it stores relevant information on each service performed, which can only be viewed and controlled by the taxi central. It also serves as a persuasive tool towards potential criminals by making them avoid committing any criminal act, knowing that the means of transport is monitored by the security system.

This device, in addition to providing security to the system, provides control of the services that a taxi driver performs periodically since their information is stored in a database on a web server and can be analyzed by the taxi central.

The creation of the rest API is a very important implementation for any system since HTTP requests can be made from the server to the client or vice versa regardless of the programming language used. In this case, for example, requests are made from JAVA (Angular and Android) and Python using a single control structure to perform data exchange through PHP in Laravel.

The use of the Raspberry Pi 4 facilitated the development of the prototype since it was not necessary to design any hardware for the acquisition and processing of the image of the interior of the taxi. However, when compiling the code, a slowdown of approximately three seconds was obtained for updating each frame, even using threads in the algorithm. If this methodology had not been used, the slowdown of the system would increase to approximately eleven seconds, a time that is critical and would cause serious inconveniences, making this an inefficient system.

Through the use of a convolutional neural network, it was possible to efficiently and in real time detect the faces of the users and the taxi driver, thus achieving the automation of the activation of the service performed, based on the number of people detected inside the taxi. However, in night conditions when a car or motorcycle reflects light permanently through the rear windshield of the taxi, at some angles it is difficult to



detect the faces of the passengers behind as this reflection distorts the image capture.

The mobile application allowed the user to graphically show the basic information of the taxi driver and the taxi, as well as the map where the route that the taxi is making is dynamically displayed in real time. With this information, the user can verify that the driver of the taxi is indeed the taxi driver authorized and registered by the taxi central, providing greater confidence to the user.

The website allows the taxi center to access the information on the services in a simple and visually pleasing way. You can consult a specific service based on the search filters provided in the table where your information is displayed, thus facilitating the search when the system contains too many records.

The prototype depends significantly on its internet connection, although the mobile device has a 4G connection, in some areas this connectivity can be reduced due to lack of coverage, which can cause a delay in the route drawing and the upload of images to the web server.

4. RECOMMENDATIONS

Although the project is ambitious, some improvements can be made to optimize the prototype to obtain a much more structured product, such as:

Since there is a slowdown in the update of each frame of the algorithm implemented in the Raspberry Pi 4, functional tests can be carried out with boards such as Coral from Google or Jetson Nano from NVIDIA since they promise better performance when designed for work with artificial intelligence.

It is recommended to look for a camera with better specifications such as resolution and autofocus to improve the quality of photos stored on the server.

In the same way, you could work with a specialized camera for passengers in the front area and another for passengers in the rear area to have better quality of the photographs. By doing this, an added value can be given to the system since the second camera can visualize all the rear seats, allowing to capture an object that the user leaves abandoned in this area since with a single camera only the central area of these seats because the front seats make an obstruction.

The implementation of 5G coverage in the country would make it possible to replace the capture of images with a video transmission at all times so that the taxi center could monitor the taxi drivers and establish communication by video call if required.

When going from a prototype to a final security system, the embedded system must be located in a place where it is difficult to access to avoid manipulation.

To start up the security system, it is necessary to comply with the regulation of video surveillance systems imposed by the superintendency of industry and commerce SIC, since there are framed the duties and responsibilities that the system must comply with when dealing with the manipulation of personal data.

REFERENCES

- Anaconda Navigator. S. F. Anaconda Navigator (Versión 1.6.9). Entorno de desarrollo Python. Consultado: 12 de Febrero de 2021. <https://www.anaconda.com/products/individual>
- Android Fashion Challenge. Tema ¿Qué Es Android. Consultado: 07 de Enero de 2021. Disponible en: <https://sites.google.com/site/androidfashionmx/home/tema>
- Angular. S. F. Angular. Versión 7.3.10. Framework de desarrollo web. Consultado: 10 de Junio de 2021. <https://angular.io/>
- Angular. Introduction to the Angular Docs. Introducción a Angular Docs. Consultado: 8 de Junio de 2021. Disponible en: <https://angular.io/docs>
- Apache Friends. S. F. Xampp. Distribución Apache. Consultado: 21 de Julio de 2021. <https://www.apachefriends.org/es/index.html>
- Baquero, J. ¿Qué es Laravel?. 11 de Diciembre de 2015. [Consultado: 13 de Enero de 2021. Disponible en: <https://www.arsys.es/blog/programacion/que-es-laravel/>
- Ceballos, J. Curso Python. Volumen XX: Hilos (Threading). Parte I. 13 de 07 de 2017. Consultado: 08 de Enero de 2021. Disponible en: <https://www.redeszone.net/2017/07/13/curso-python-volumen-xx-hilos-parte-i/>
- Composer. S. F. Composer. Versión 1.10.13. Manejador de dependencias PHP. [Consultado: 11 de Julio de 2020. <https://getcomposer.org/>
- Developers Android. Android Studio. Versión 3.5.1. Entorno de desarrollo Android. Consultado: 21 de Febrero de 2021. Disponible en: https://developer.android.com/studio/?gclid=CjwKCAjw2dD7BRASEiwAWCtCb5JGYEK4Y7KAYZQTDBcUDLE_xdsgsc7P5ZS1uFOx4BHAV34A8HR5PB0CRjUQAvD_BwE&gclid=aw.ds
- Developers Android. Documentación para desarrolladores de apps. [Consultado: 16 de Octubre de 2020]. Disponible en: <https://developer.android.com/docs>
- DevExpress. S. F. DevExtreme. Suite de componentes HTML y JavaScript. Consultado: 24 de Octubre de 2020]. <https://js.devexpress.com/>
- ELP. Cámara web ELP 1080P. Consultado: 10 de Junio de 2021. Disponible en [Fotografía]: <http://www.elpcctv.com/>
- Himashaharinda. Face-detection-app. Consultado: 25 de noviembre de 2020. Disponible en GitHub: <https://github.com/HimashaHarinda/face-detection-app>



Hipertextual. 27 de Junio de 2020. Consultado: 14 de Enero de 2021. Disponible en: <https://hipertextual.com/archivo/2014/05/que-es-api/>

Iconfinder. S. F. Map Marker. Ícono. Consultado: 17 de Enero de 2021. Disponible en: <https://cdn0.iconfinder.com/data/icons/small-n-flat/24/678111-map-marker-32.png>

Iconfinder. S. F. Sport Finish. ícono. Consultado: 17 de Enero de 2021. Disponible en: <https://cdn1.iconfinder.com/data/icons/smallicons-sport/32/finish-32.png>

Laravel. S.F. Laravel. Versión 5.7.29. Framework de desarrollo web. Consultado: 15 de Agosto de 2020. <https://laravel.com/>

Laravel. S. F. The PHP Framework for Web Artisans. El marco PHP para artesanos web. Consultado: 3 de Junio de 2021. Disponible en: <https://laravel.com/>

Merino, M. Google lanza al mercado Coral, la placa de 150 dólares para desarrolladores de inteligencia artificial en edge computing. 23 de Agosto de 2020. Consultado: 3 de Enero de 2021. Disponible en: <https://www.xataka.com/inteligencia-artificial/google-lanza-al-mercado-coral-placa-150-dolares-para-desarrolladores-inteligencia-artificial-edge-computing>

Metronic. S.F. Metronic Plantilla. Consultado: 13 de Febrero de 2021. <https://keentthemes.com/metronic/>

Najam, R Syed. Multithreading with OpenCV-Python to improve video processing performance. Multihilos con OpenCV-Python para mejorar el rendimiento del procesamiento de video. 5 de Julio de 2018. [Consultado: 22 de Junio de 2021. Disponible en: <https://nrsyed.com/2018/07/05/multithreading-with-opencv-python-to-improve-video-processing-performance/>

Node JS. S. F. Node JS. Versión 13.7.0. Entorno de ejecución para JavaScript. Consultado: 19 de Julio de 2020. <https://nodejs.org/es/>

Nvidia. Aprovecha El Poder De La Ia De Nvidia. Consultado: 13 de Junio de 2021. Disponible en: <https://www.nvidia.com/es-la/autonomous-machines/embedded-systems/jetson-nano-developer-kit/>

Nvidia. OpenCV. Consultado: 29 de Agosto de 2020. Disponible en: <https://developer.nvidia.com/opencv>
O.S.Group. Que es un servidor web. Consultado: 08 de Enero de 2021. Disponible en: <https://www.osgroup.co/que-es-un-servidor-web/>

Opencv. OpenCV deep learning module samples [Mestras de aprendizaje profundo OpenCV. Consultado: 22 de Octubre de 2020. Disponible en: <https://github.com/opencv/opencv/tree/master/samples/dnn>

PDF-Manuales. Curso Libro Laravel 5 Conceptos básicos y ejemplos en pdf. 19 de Febrero de 2018. Consultado: 14 de Junio de 2021. Disponible en: <https://www.pdf-manual.es/programacion-web/php/175-libro-laravel-5-conceptos-basicos-y-ejemplos.html>

Pérez, M. Firebase, qué es y para qué sirve la plataforma de Google. 14 de Octubre de 2016. Consultado: 07 de Enero de 2021. Disponible en: <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/#comments>

Placeholder. S. F. Placeholder image. Imagen. Consultado: 17 de Enero de 2021 Disponible en: <http://placeholder.it/180>

Python. Python. Versión 3.6.8. Lenguaje de programación. 24 de Diciembre de 2018. Consultado: 03 de Febreo de 2021] <https://www.python.org/downloads/release/python-368/>

Quality Devs. ¿Qué es Angular y para qué sirve? 16 de Septiembre de 2020. Consultado: 05 de Enero de 2021. Disponible en: <https://www.qualitydevs.com/2020/09/16/que-es-angular-y-para-que-sirve/>

Semanticwebbuilder. Sistemas Embebidos: Innovando hacia los Sistemas Inteligentes. Consultado: 07 de Enero de 2021. Disponible en: http://www.semanticwebbuilder.org.mx/es_mx/swb/Sistemas_Embebidos_Innovando_hacia_los_Sistemas_Inteligentes_

Silva, S. Y. Freire E. 2021. Esquema funcionamiento red neuronal convolucional. 23 de Noviembre de 2020. Esquema. Consultado: 06 de Enero de 2021. Disponible en: <https://medium.com/@bootcampai/redes-neuronales-convolucionales-5e0ce960caf8>

Silva, S. Y Freire E. 2021. Intro a las redes neuronales convolucionales. 23 de Noviembre de 2020. Consultado: 06 de Enero de 2021. Disponible en: <https://medium.com/@bootcampai/redes-neuronales-convolucionales-5e0ce960caf8>

Sitiobigdata.com. OpenCV Python: Face Detection Neural Network. OpenCV Python: Red neuronal de detección de rostros. Consultado: 9 de Octubre de 2020. Disponible en: <https://sitiobigdata.com/2020/05/01/opencv-python-face-detection-neural-network/>

Superintendencia de Industria y Comercio. Porección de datos personales en sistemas de videovigilancia. Consultado: 23 de Junio de 2021. Disponible en: https://www.sic.gov.co/sites/default/files/files/Nuestra_Entidad/Guia_Vigilancia_sept16_2016.pdf



Tic. Portal. Base de datos. 09 de Julio de 2020.
Consultado: 08 de Enero de 2021. Disponible en:
<https://www.ticportal.es/glosario-tic/base-datos-database>

Towards data science. Face Detection Models: Which to
Use and Why?. Modelos de detección facial: ¿Cuáles usar
y por qué?. 02 de Julio de 2020. Consultado: 8 de Octubre
de 2020. Disponible en:
<https://towardsdatascience.com/face-detection-models-which-to-use-and-why-d263e82c302c>